D7.5 Annual Report on Collaboration - Year 2

Grant agreement no.	780785
Project acronym	OFERA (micro-ROS)
Project full title	Open Framework for Embedded Robot Applications
Deliverable number	D7.5
Deliverable name	Annual Report on Collaboration - Year 2
Date	December 2019
Dissemination level	Public
Work package and task	7.2
Author	FIWARE
Contributors	eProsima, Bosch, PIAP
Keywords	micro-ROS, robotics, ROS, microcontrollers, collaboration
Abstract	This annual report on collaboration activities performed under the umbrella of the OFERA project and in relation to the micro-ROS platform.

Abbreviations

Term	**Definition**
DDS	Data Distribution Service
DDS XRCE	DDS for extremely resource-constrained environments
IMU	inertial measurement unit
MCU	microcontroller
rcl	ROS 2 client support library
rmw	ROS 2 middleware interface
ROS	Robot Operating System
RTOS	real-time operating system

1 Introduction

In the first year of the project, not much progress on collaboration could be done as the project was still on an early stage. The focus was put in initiating collaboration activities:

- with other EU-funded projects in the Robotics area
- with the FIWARE Community
- with the ROS Community

In this reporting period, the collaborations initiated along these three axes have been strengthened. In particular, the collaboration with EU projects such as ROSIN and RobMoSys are very promising and they are already giving results in the form of concrete specifications and software contributions. In this line, new collaborations are being explored within the framework of another European project, the DIH² Project. During 2020, this project will finance different experiments where robotic systems will be applied in manufacturing environments to implement agile production concepts and we are trying to identify use cases where micro-ROS can be adopted. The opportunities for collaboration will depend on the projects that are finally selected for funding.

Regarding FIWARE, the progress of the Micro XRCE-DDS component, one of the results linked to micro-ROS, continues according to its roadmap as incubated FIWARE GE. A regular report of this progress is provided and followed up at the level of the FIWARE Technical Steering Committee (TSC) where overall technical direction of the FIWARE Open Source Community is driven. Furthermore, micro-ROS is linked to the activities of the FIWARE Chapter focused on interfaces with IoT, Robotics and 3rd-party systems through the XRCE-DDS and FastRTPS components. Last but not least, within the framework of FIWARE Technical Roadmap activities, a specific working group has started to work on the design of collaborations focused on the robotic sector. In it, members from the eProsima and FIWARE technical teams collaborate with members from other organizations like Atos, Fraunhofer IML and the Japanese TIS Inc. The evolution of this working group and the continued presence of micro-ROS in it through FIWARE and eProsima contributions makes micro-ROS accessible to new organizations and, therefore, sets a promising context for initiating new collaborations.

On its turn, the close relationship between micro-ROS and the ROS community is materialized through different lines of collaboration, most of them under the umbrella of ROS2 activities. The presence of two micro-ROS partners, Bosch and eProsima, in the ROS Technical Steering Committee is particularly noteworthy. Both initiated the formation of an Embedded Special Interest Group (SIG) for ROS 2 during the ROSCon event in 2018 and, nicely, this Embedded SIG has been turned into an official ROS 2 Working Group (WG) by the ROS TSC. Furthermore, Bosch also participates in ROS 2 Real-Time WG, which significantly increases the presence, visibility and collaboration opportunities of micro-ROS within this community. These WGs have regular meetings on a monthly basis, some bilateral cooperations with organizations like Robotics, RENESAS or Nobleo have been initiated and the project will continue to engage the ROS community as well as other European partners within the ROS 2 embedded WG. To that aim, project status updates, as well as interesting news, will be posted not only to the micro-ROS site but also to the ROS Discourse embedded category.

2 Collaboration with other EU-funded Projects in Robotics

2.1 Collaboration meeting with ROSIN and RobMoSys

The OFERA project review was held in Luxembourg to coincide with other H2020 project reviews that were scheduled in the same place and similar dates. RobMoSys and ROSIN organised a workshop meeting with some participants from the different European H2020 project consortia and OFERA joined this workshop.

The meeting took place on September the 12th in Luxembourg. In this meeting each project presented their experiences, current status and outcomes. This meeting provided a good opportunity to get updates on co-existing EU funded projects working in similar target domain. This workshop was a good frame to interchange experiences on software engineering, Robotics, European Funds and the future of Robotics within Europe Industry and to create new contacts within partners from other Consortiums.

Specifically with RobMosys and ROSIN, as explained further later RobMoSys, MROS has some synergies with work done for OFERA. And ROSIN has already been a good source of FTP for OFERA consortium members adding value to the Robotics ecosystem in Europe.

With ROSIN there has been good and fluent collaboration in the past and further collaboration is envisioned, not only with FTPs but with the OFERA consortium members being invited to ROSIN conferences as, for example, the ROS-Industrial EU Tech Workshop held in Stuttgart in May 2019 and the invitation to participate in ROS-Industrial Conference in Stuttgart in december 2019 where eProsima and BOSCH gave respective talks.

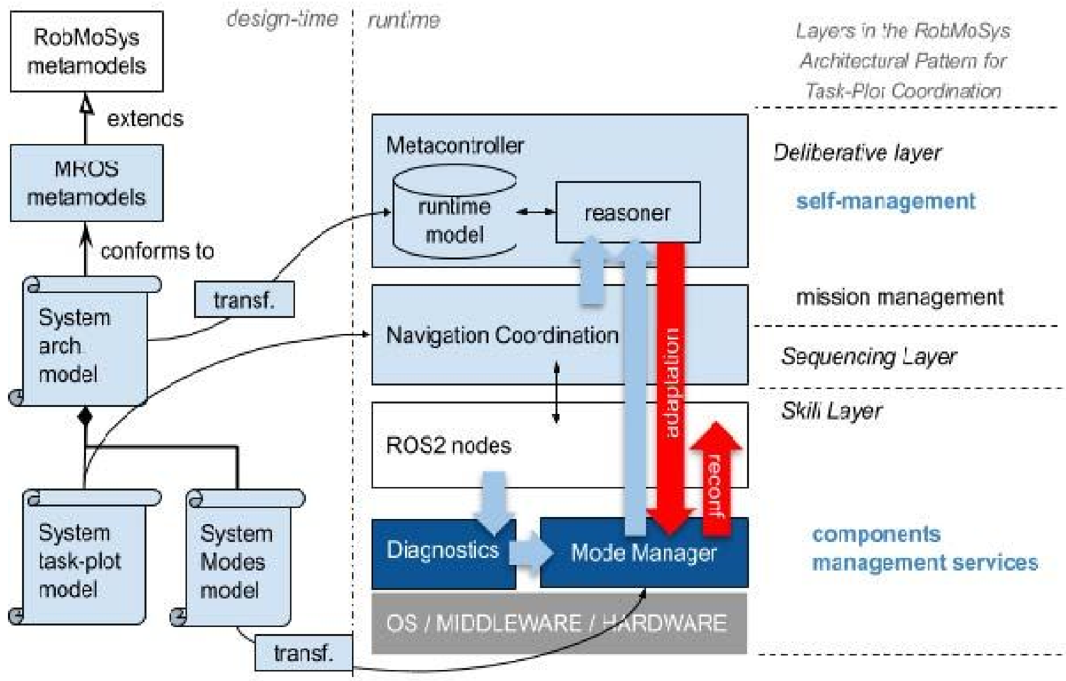
2.2 ITP MROS in RobMoSys

The EU project RobMoSys envisions an integrated approach for robotic software platforms by applying model-driven methods and tools and top of code-centric platforms like ROS. The model-based approach for runtime system (re-)configuration by System Modes, developed in Task 4.3 of OFERA, can be considered as one contribution towards this vision.

An academic consortium by TU Delft, Universidad Politécnica de Madrid, Universidad Rey Juan Carlos in Madrid, and IT University Copenhagen approached Bosch Research for joining forces towards the development of a model-based approach for even more advanced reconfiguration and metacontrol methods based on architectural models, system modes, and ontological reasoning.

Together, the five partners proposed an Integrated Technical Project (ITP) named Metacontrol for ROS 2 (MROS) in the second open call of RobMoSys for this endeavor. MROS was accepted and launched successfully together with ten other ITPs in a joint kick-off event in Munich in October 2019. Two members from the OFERA consortium participated for Bosch in this event.

The following architectural diagram provides an overview of the elements to be provided by MROS (in light blue), including the integration of micro-ROS System Modes (dark blue elements).



The research and development on metacontrol will be conducted by example of two mobile robotics pilots running the ROS 2 Navigation stack. One major goal of MROS is to bring the system modes concept mainline in the ROS 2 Navigation stack, which would be a great success in the dissemination of the this result from micro-ROS.

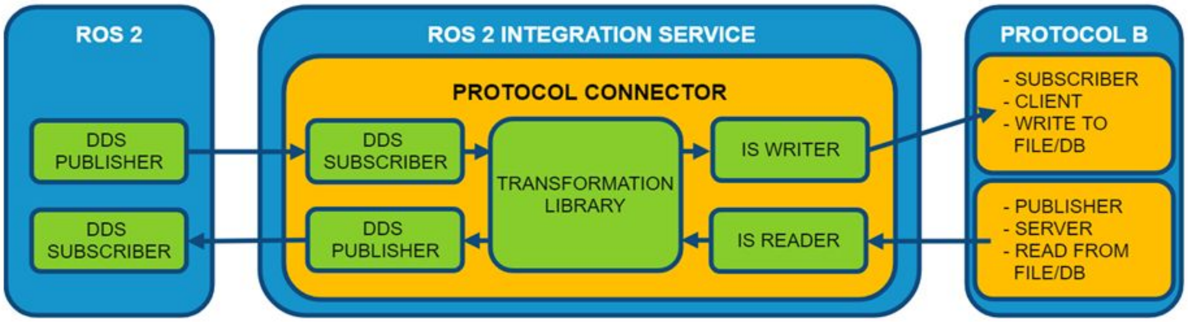
The MROS website can be found at https://robmosys.eu/mros/. A blog post by RobMoSys on the kick-off event can be found at https://robmosys.eu/robmosys-kicking-off-11-new-itps/.

2.3 SOSS developed in FTP ROSIS with OSRF

The main goal is offering a fully open source solution for two general ROS2 use cases:

1.- Integration: Connect ROS2 to other protocols (MQTT, zeroMQ, a file/DB, etc): There are some existing bridges, such as ROS2 to ROS1 Bridge, or FIROS2 (FIWARE to ROS2), but this project will offer a common framework, standard interfaces, and useful services to create general Bridges and transformations.

2.- Routing: Enable ROS2 over Internet/WAN. Currently, ROS2 just allow LAN comms because of the underlying protocol of ROS2. The project will offer routing services for ROS2 traffic in several scenarios.



Public Summary of Milestone 1:

ROS2 Integration Service provides a complete tool to integrate other technologies with ROS2 easily and enable ROS2 on WAN/Internet. In this initial release, the core functionality of ROS Integration Services implementation was completed. This first release:

- It is fully integrable with Colcon . ROS Integrations Services integrates into ROS 2 workspaces as a package.
- The configuration is done using YAML format which increases the readability compared to XML configurations.
- Provided a set of Docker images, so ROS Integration Services is ready to be used out of the box.
- Define and provide a well-defined API to create new system integrations.
- Developed Bridges: FIWARE , DDS, ROS2, ROS1, WebSocket, REST.
- Provide usage examples and tests: TCP tunnel using DDS, ROS2 domain change and a connection ROS2 – FIWARE .
- Provides user documentation.

This initial release uses SOSS (System Of Systems Synthesizer) as core communications enabler. SOSS provides a set of tools and configuration utilities to communicate different communication middlewares.

The outcome of this first Milestone can be found in a set of publicly available repositories:

- ROS-Integration-Service: Central repository for ROSIS. Uses SOSS as core architecture: https://github.com/eProsima/ROS2-Integration-Service
- SOSS-FIWARE: Integration of FIWARE with any other SOSS plugin: https://github.com/eProsima/SOSS-FIWARE
- SOSS-DDS: Integration of native DDS (Fast RTPS used as implementation): https://github.com/eProsima/SOSS-DDS
- soss_v2: Configuration and libraries loader. Includes set of system handlers/bridges: https://github.com/osrf/soss_v2

3 Collaboration with the FIWARE Community

FIWARE brings a curated framework of open source software platform components (referred to as FIWARE Generic Enablers - GEs) which can be assembled together and with other third-party components to build platforms that support the development of Smart Solutions faster, easier and cheaper. A more detailed technical description of FIWARE can be found on the FIWARE website or GitHub.

The FIWARE NGSI API provides a simple yet powerful API for solving a basic need in any smart solution: how to gather, manage and provide access to context information. There is a core Context Broker component in every "Powered by FIWARE" platform which supports this API and enables the integration of the rest of platform components. The key contribution of the harmonised FIWARE NGSI API and the central Context Broker is the provision of a basis for interoperability of smart solutions/services that run on top of "Powered by FIWARE" platforms as well as enabling their portability (replicability) across different "Powered by FIWARE" platforms.

A rich suite of complementary FIWARE components can be combined with the FIWARE Context Broker in a modular architecture and integrated as part of a "Powered by FIWARE" platform. These components are referred to as FIWARE Generic Enablers (GEs). The complete set of FIWARE GEs are structured in several FIWARE chapters (see Figure 3.1):

- *Core Context Management chapter*, comprising the core FIWARE Context Broker component as well as components enabling integration with multiple alternative data sinks for storage and further processing of historic context data
- Chapter comprising FIWARE GEs helping to implement the *interface with IoT devices, robots and third-party systems*, capturing updates on context information and translating required actuations;
- Chapter comprising FIWARE GEs for *advanced monitoring, using dashboard and analytical support tools, as well as processing and analyzing current and historic context data* using event rules, advanced Big Data and AI algorithms, targeted to support smart decisions or the smart automation of processes;
- Chapter comprising FIWARE GEs dealing with *management, publication and monetization of context data and services*, preserving defined access and usage control policies.

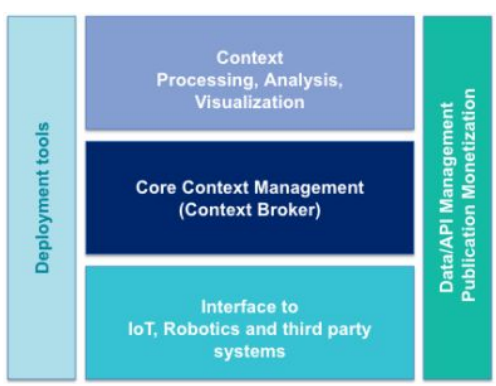


Figure 3.1 FIWARE components are organized in chapters and are referred to as FIWARE Enablers

3.1 Integration of micro-ROS results in FIWARE

Micro XRCE-DDS technology, has been contributed to FIWARE as Incubated FIWARE GE to the collection of enablers focused on interfacing with IoT devices, robotics and 3rd party systems. An Incubated FIWARE GE is an open source software product whose owner has proposed for adoption as a new FIWARE Generic Enabler (i.e., component) of the FIWARE platform framework. Consideration as first-class FIWARE GE is achieved once well established requirements for FIWARE contribution are met (the specification of these requirements is public, it can be accessed through this link).

The application process of the Micro XRCE-DDS technology to become a FIWARE Enabler started on February 18 (2019). On the 25th of February, the application was discussed and approved by the FIWARE TSC. After that, Micro XRCE-DDS was incorporated to the FIWARE Release 7.6 and has continued since then. It was presented for the first time as part of the FIWARE Summit events in Genoa (21-22 May, 2019). The entry point within the FIWARE catalogue to get familiar with the Micro XRCE-DDS technology is this one (a snapshot of this site is attached below, Figure 3.1.1).

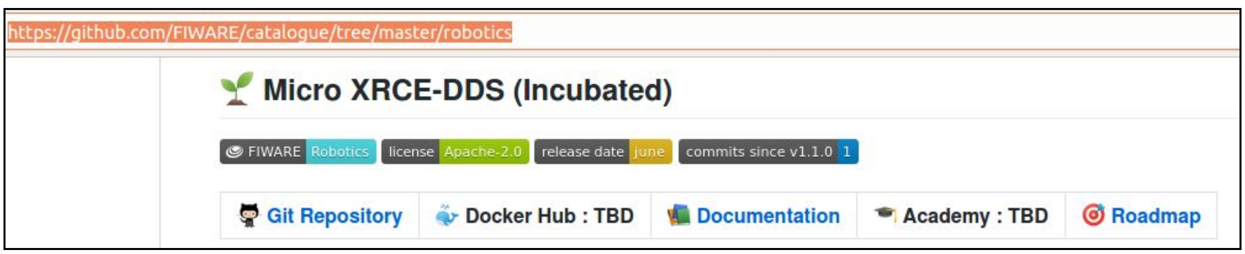


Figure 3.1.1 The Micro XRCE-DDS component has a dedicated space within the FIWARE Catalogue

3.2 Participation in FIWARE Technical Roadmap activities

Since the Micro XRCE-DDS (product that results from micro-ROS activities) and Fast RTPS (default middleware in ROS2) have been adopted as FIWARE enablers their status is reported on a Bi-monthly basis at the FIWARE's TSC level. Besides, these sessions are not limited to status reports, they also include the discussion on the roadmap of these components roadmap as well as on the opportunities for starting new collaborations and/or promoting their use in scenarios where the FIWARE community plays a role. In particular, the integration of Micro XRCE-DDS into FIWARE has helped to make this component visible in projects (EU funded or not) where FIWARE is being applied in connection to robotics (e.g., the DIH² Project).

On the one hand, robotics is one of the fields that has been identified as particularly relevant to the expansion of FIWARE. On the other hand, this close collaboration with micro-ROS creates opportunities for other results in this project to be added to the list of FIWARE enablers and, hence, disseminated through its extensive community. Thus, in addition to including micro-ROS aspects at the TSC level, a specific FIWARE working group on robotics has been set up and some strategic collaboration opportunities have already emerged. For example, with the Japanese company TIS, which has become a member of the FIWARE community and has brought in robotic use cases to be addressed at the WG.

4 Collaboration with the ROS Community

4.1 Contributions to ROS 2 Core Packages

A key element in ensuring the long-term maintenance of micro-ROS is bringing as many features and changes for micro-ROS as possible mainline into the ROS 2 core packages at https://github.com/ros2/. Corresponding pull requests have to be coordinated with the community very thoroughly as they affect all ROS developers. Most pull requests are discussed over several weeks before they are accepted. The Open Robotics Software Foundations also runs extensive tests on these pull requests. In the last to years, we contributed to the two main repositories rcl and rclcpp as well as to several middleware-related repositories.

In general, we try to prepare such pull requests by discussions in the ROS 2 Technical Steering Committee and the Embedded or Real-Time Working Group, as explained below. Also, we raised a public discussion on the design of ROS for deeply embedded systems by a pull request to https://github.com/ros2/design/, which triggered almost 100 comments from the community.

4.2 Interaction with ROS Developers

One of the important points in the success of ROS as Robotic framework is the size and implication of its community of developers. The OFERA consortium knows that a good interaction with this ROS developers community plays a vital role in the success of micro-ROS and in its future. OFERA consortium interacts directly with ROS developers in different ways.

A. OFERA members are active in existing ROS developers oriented communication channels
    - In the official ROS discussion forum https://discourse.ros.org
    - and in https://answers.ros.org/questions/ an stackoverflow-like website, which recently reached 50.000 questions.

B. OFERA consortium members are also active contributors to ROS GitHub repositories. This collaboration is done via pull request and issues in which discussions and design are open to the community, including OFERA members.

C. Aside from the previous “official” ROS channels, OFERA has launched a micro-ROS dedicated Slack workspace: https://micro-ros.slack.com/.
    - The slack has by the date of this report 97 members, with 2698 messages spread in 4 open channels (OFERA consortium has 2 private channels dedicated to project matters).

D. A fourth way of interaction with the community of ROS developers is the participance of Consortium members in community oriented events like: ROSCon, ROS-Industrial Conferences and Workshops… more detail can be found in *D7.2 Annual Report on Communication and Dissemination - Year 2.*

4.3 ROS 2 Technical Steering Committee

The ROS 2 Technical Steering Committee (TSC) was initiated by the Open Source Robotics Foundation (OSRF), i.e. the foundation behind ROS, in September 2018 to broaden participation to accelerate ROS 2 delivery. The TSC defines the roadmap for the development of ROS 2 and the members each contribute to the development of core libraries and tools. Bosch was one of the founding members of the ROS 2 TSC. Later, eProsima and Acutronic Robotics from the OFERA consortium joined the TSC.

Other TSC members are Amazon, Apex.AI, Canonical, Intel, LG Electronics, Microsoft, ROBOTIS, Samsung, GVSC, Tier IV, and the Toyota Research Institute (as of 20 September 2019). The list of ROS 2 TSC members is provided at https://index.ros.org/doc/ros2/Governance/.

The TSC membership of multiple OFERA partners allows to align the micro-ROS developments well with the roadmap for the next ROS 2 release. In addition, it is a great opportunity to stay at the forefront in technical discussions about the ROS 2 core layers.

4.4 ROS 2 Working Groups

The TSC aims at conducting these technical discussions as publicly as possible. That is why the TSC charters make it possible to establish *Working Groups* (WG) to discuss specific topics in greater detail. Currently, there are six WGs, on embedded systems, navigation, real-time, safety, security, and tooling.

The list of WGs is also provided at https://index.ros.org/doc/ros2/Governance/, just as the list of TSC members. The OFERA consortium is involved in two of these working groups.

4.4.1 Embedded Systems WG

Last year OFERA Consortium started a Special Interest Group in ROS 2 in embedded devices during ROSCon2018. During the second year of the project regular meetings have been scheduled by Consortium members. There have been 5 total meetings attracting the community from different points of the planet.

As a remarkable milestone in this community oriented action, the SIG was included in the ROS 2 governance schema as an official Working group under the name of Embedded Systems Working Group. Official list of WF can be found at https://index.ros.org/doc/ros2/Governance/.

The working group is in early stages in community collaborations terms, however already have been interesting meetings in which use cases for embedded devices within Robotics and ROS 2 in embedded devices has been discussed.

All minutes of the WG meetings can be found at https://discourse.ros.org/ by searching for “ROS 2 Embedded WG” or previously “ROS 2 Embedded SIG”. From now on the OSRF assigned tag: “wg-embedded” will be the recommended search keyword.

4.4.2 Real-Time WG

The Real-Time WG has been initiated by the TSC in April 2019 to concentrate existing efforts towards ROS 2 real-time behavior. From the OFERA consortium, Bosch and eProsima participate in this WG since its inception. In the initial announcement on this WG, the works on the Callback-group-level Executor in the context of OFERA were mentioned explicitly as one starting point.

While real-time safety has been an objective of ROS 2 since the beginning, a lack of capacity meant that checking compliance, and looking at parts of the stack below and above the core framework, had not so far been addressed with the necessary rigor. Therefore, in the first meetings, the WG focused on analyzing the current state of the ROS 2 stack. Several organizations developed and contributed tools to analyze/test/benchmark the ROS 2 stack in this regard. On the part of OFERA, tooling for runtime tracing of ROS 2 systems and benchmarking of ROS 2 Executor implementations has been contributed (cf. deliverables D4.5 and D5.5).

In the meanwhile, based on the results, some first improvements have been developed by members of the Real-Time WG. For example, a new static, runtime-efficient Executor for the C++ layer (rclcpp) was developed by the Dutch company Nobleo Technology, after a joint benchmarking effort with Bosch. In addition, first concepts for deterministic execution with ROS 2 have been drafted and discussed.

All minutes of the WG meetings can be found at https://discourse.ros.org/ by searching for “real-time working group”.

4.5 Individual Open-Source Cooperations

4.5.1 Executor Benchmarking Suite

In a bilateral open-source cooperation, Nobleo Technology from Eindhoven (The Netherlands) and Bosch developed a small tool suite for benchmarking of ROS 2 Executor implementations. The tool suite has been published at github.com/nobleo/ros2_performance/ and github.com/micro-ROS/rcl_executor/tree/feature/testbench.

Together with the runtime tracing tools developed in OFERA for ROS 2 and micro-ROS, it has been used to identify performance issues in the default rclcpp Executor as well as to evaluate the performance of the LET Executor for micro-ROS.
On the basis of the analysis results on the default rclcpp Executor, Nobleo Technology even developed an improved, static version executor https://github.com/nobleo/static_executor and contributed it to the core rclcpp repo https://github.com/ros2/rclcpp/pull/873.

4.5.2 Crazyflie

Crazyflie is the robotic platform used as micro-ROS drone community demo. This platform has its firmware publicly available as open-source. In this sense and in order to achieve a working micro-ROS application in Crazyflie quadcopter, OFERA members needed to make improvements and changes in the existen firmware code. These improvements have been offered in the form of pull request to the firmware repository owners. Bitcraze, the firmware developers have happily included micro-ROS proposed changes in their master branch and will be issued in their next release.

micro-ROS provided different kinds of improvements to the existing firmware:

1. Improved repository layout and dependency management. This was required to support the latest version of FreeRTOS in the firmware required by micro-ROS FreeRTOS port.
2. Allow memory control defined from out of tree applications. This will improve real time support at application level.
3. Fix corrs compilation issues using a custom toolchain. This allows Crazyflie firmware to be built with external build tools.

Also OFERA members keep active in issues and discussions regarding Crazyfly platform build system and applications support.

4.6 Other collaborations

As it was set in the objectives of the MS7 (Customer Pilot I) other collaborations have been started regarding the identification/definition of potential use cases to implement customer pilots. The first of these collaborations is being carried out with the Renesas Electronics Corporation, the second one with drone solutions regarding the PX4 autopilot.

In the collaboration with Renesas, the potential application of micro-ROS is connected with one of their components the RX65N microcontroller. In their use case, two boards that are based on this microcontroller must interact, one of them being in charge of controlling the sensing of a robotics system while the other is responsible for the control of its actuators. The use case gravitates around the solution that was presented in this post.

Regarding the collaboration with Dronesolutions.io, it considers micro-ROS to define a distributed architecture that builds upon a DDS domain, making a clear separation between the features that run in processors outside the drone and those that drone runs onboard. By relying on micro-ROS, the use case aims to implement this link and facilitate the integration of additional ROS2 features that will bring added value into the existing platform. The use case gravitates around the solution presented in this document.